SWPY005—May 2002
White Paper

Bringing Streaming Video to Wireless Handheld Devices

By Sebastein De-Gregorio, Audio/Video Project Lead
Madhukar Budagavi, Member Group of the Technical Staff
Jamil Chaoui, WW OMAP Software Application Manager
Texas Instruments Wireless Terminals Business Unit

Introduction

Conventional wisdom has it that communication bandwidth will limit the viability of streaming video applications on wireless terminal devices like cellular telephones, smartphones, personal digital assistants (PDAs) and even compact laptop computers. Certainly bandwidth is a factor, but not by a long shot is it the only factor that will affect wireless streaming video.

In fact, video compression techniques like MPEG-4 and others are reducing the wireless bandwidth needed for streaming video. But there still remain other factors, which are just as critical as bandwidth, if not more so. Beyond bandwidth, the other half of the streaming video equation is the architecture and capabilities of the terminal device itself. Even if unlimited wireless bandwidth were suddenly to become available, service providers and terminal device manufacturers would have no assurance they could provide a high-quality streaming video experience, unless the architecture of the terminal device featured a solid foundation for the application and the processing capabilities to pull it off.

The fact of the matter is that the wireless communications environment is not readily conducive to streaming video. On one hand, high-quality streaming video is no easy task. It involves downloading, decoding and playing video and audio simultaneously, and with no or very limited re-buffering taking place. On the other hand, the wireless environment is fraught with challenges like interference, multipath fading, bit stream errors and mobile terminal devices that are moving targets, darting in and out of areas that may have different transmission speeds as well as other characteristics. To be able to cope with the demands of a streaming video application and the vagaries of wireless communication in general, a terminal device must have a unique blend of hardware and software especially well suited to wireless streaming video.

Established Architecture

The vast majority of today's voice-only, second-generation (2G) wireless communications devices are based on a dual-processor architecture featuring a digital signal processor (DSP) and a general-purpose processor. In this architecture, which was pioneered by Texas Instruments, the DSP handles many of the communications tasks, like modulating and demodulating the bit stream, coding and decoding, encryption processes for security, and compression/decompression algorithms. The second processor is assigned general-purpose tasks, such as the user interface and the upper layers of the communications protocol stack.

Fortunately, this dual-processor architecture is very well suited to wireless streaming video. TI's OMAP™ dual-processor architecture has been significantly enhanced to process the most demanding multimedia applications, including streaming video. The OMAP platform features a low-power, programmable DSP and a powerful reduced instruction set computing (RISC) general-purpose processor. Because of the strenuous demands of applications like streaming video, a partitioning of the application's tasks between the two processors is critical for several reasons. First, the speed and throughput of the system will be optimized when tasks are assigned to the processor best suited to handle them. And second, assigning tasks to the appropriate processor will reduce the number of processor cycles required for each task, which, in turn, reduces the power drained from the battery and extends the usable life of the mobile device. Common sense dictates that if more processor cycles must be devoted to a task than are absolutely necessary, then the power needed to execute that task will be higher than it should be.

The OMAP architecture shown in Figure 1 has been optimized to provide the processing capabilities needed for demanding wireless streaming video applications and, at the same time, extend the battery life of mobile devices by consuming less power. In general, the tasks involved in a streaming video application can be divided into control and transport (CT), and media decode (MD). CT tasks include processing the real-time streaming protocol (RTSP) and the real-time protocol (RTP), which is a media transport mechanism. Because these tasks are not computationally intense, a general-purpose RISC processor is well suited to executing them. But MD tasks involve decoding the video bit stream, high quality audio decoding and other signal processing tasks. These processes are decidedly computationally intense. As a result, a high-performance, low-power DSP is a good fit for MD tasks.

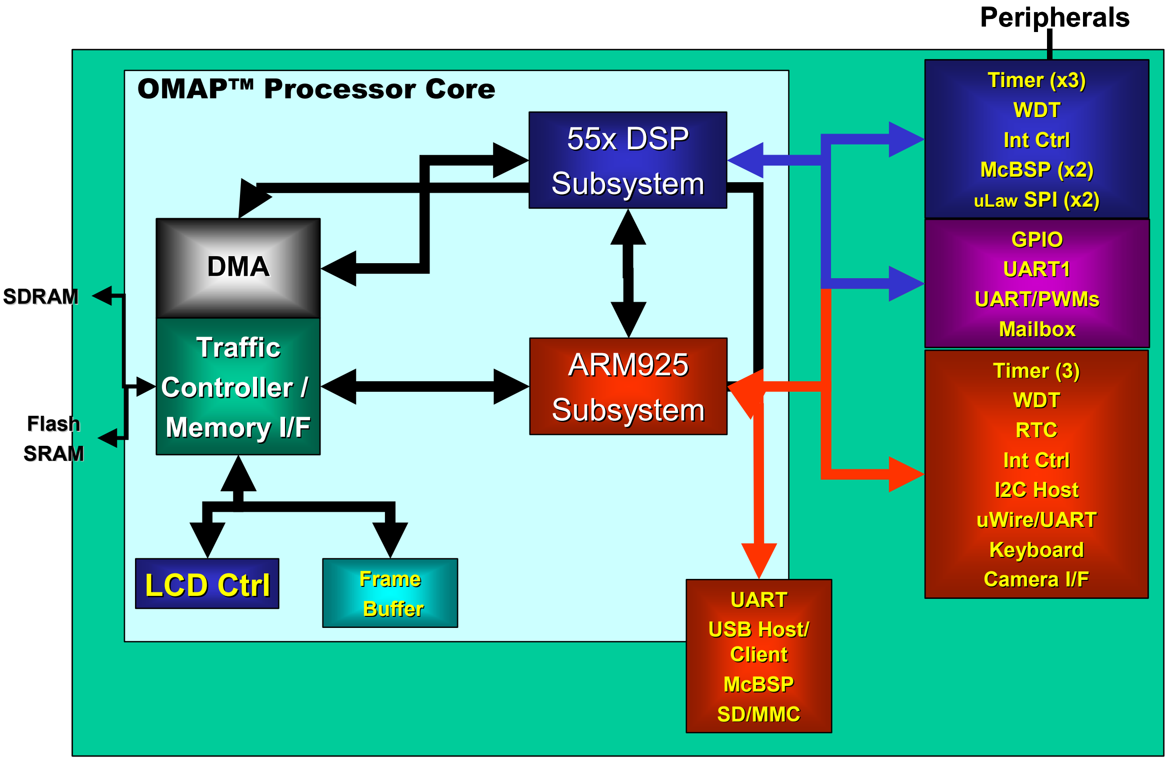

Figure 1: OMAP™ Architecture

Figure 2 shows an efficient way to map a streaming video application onto the OMAP architecture.

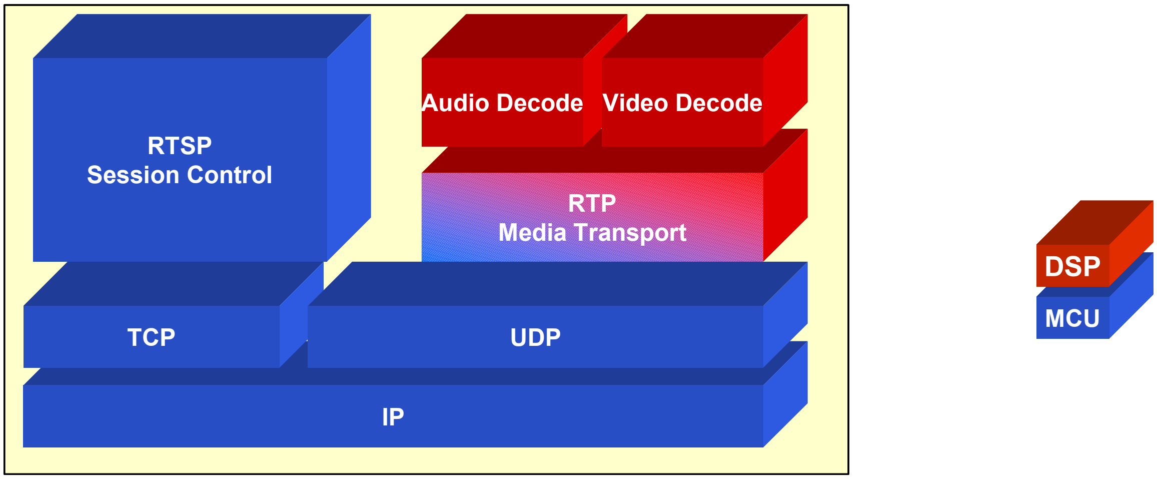

Figure 2: Partitioning of a Streaming Video Application

When a streaming video application is processing, radio signals enter the OMAP architecture by way of a modem. The general-purpose RISC processor handles the protocol (RTP/RTSP) processing and demultiplexes the audio and video data. The radio signals are then transformed into an elementary bit stream and forwarded to the DSP's internal random access memory (RAM). To minimize the processing demands on the system, streaming video applications use the current frame or image to extrapolate the following frames. A frame is moved one macroblock at a time from the video buffer into the DSP's internal RAM where it is combined with other information and sent to the display as the current frame.

The tremendous amount of data that is transferred in real-time during a streaming video application makes input/output (I/O) considerations of paramount importance. The processing capabilities of the two processors in the OMAP architecture would be wasted if data could not be moved throughout the architecture in a timely fashion. At least three direct memory access (DMA) connections are needed to avoid I/O bottlenecks, which can cripple a streaming video application. In the OMAP architecture, all of the DMA channels have access to all of the shared memory, ensuring an efficient internal data flow. One six-channel DMA connection provides access to the DSP while a second 10-channel DMA (nine generic channels and one channel dedicated to the LCD controller), the so-called system DMA, is connected to the general-purpose RISC processor. These DMA capabilities are needed to speed the movement of data structures that are typically found in streaming video applications, like two-dimensional pixel arrays, byte alignments and byte-by-byte transfers. In particular, the DSP's DMA channels are especially critical in a streaming video application because large graphic images must be quickly and constantly moved from external memory to internal memory.

Cohesiveness Is a Key

Multi-processor architectures like the OMAP platform raise the question of the cohesiveness of the processors. Conflicts between processors can arise when both processors contend for the same memory location. In addition, memory access requests initiated by either of the two processors for a certain location in memory can be processed only one at a time. A first-in-first-out algorithm for memory access requests would prove inadequate because some requests will be associated with real-time applications like streaming video and these must be guaranteed a predictable response time.

The OMAP platform is able to overcome contention between the two processors because of the traffic controller (TC), which is an inherent part of the architecture. The TC is a programmable arbitration mechanism that sits between the DSP, the general-purpose RISC processor and the OMAP platform's external interfaces. Depending on the algorithms programmed into the TC, it will prioritize memory accesses and resolve any conflicts that may arise.

TI's DSP/BIOS™ Bridge is another tool that alleviates many of these problems and assures the cohesiveness of the OMAP platform. The DSP/BIOS Bridge is a high-level software layer that overarches both the DSP and RISC processors in the OMAP architecture. At the same time though, the DSP/BIOS Bridge is a multi-threaded construct with tentacles that reach from the lowest level of device drivers to the platform's highest level. It oversees the conflicts that may arise and also serves as an abstract application programming interface (API), simplifying the software development process and reducing time-to-market for developers (Figure 3).

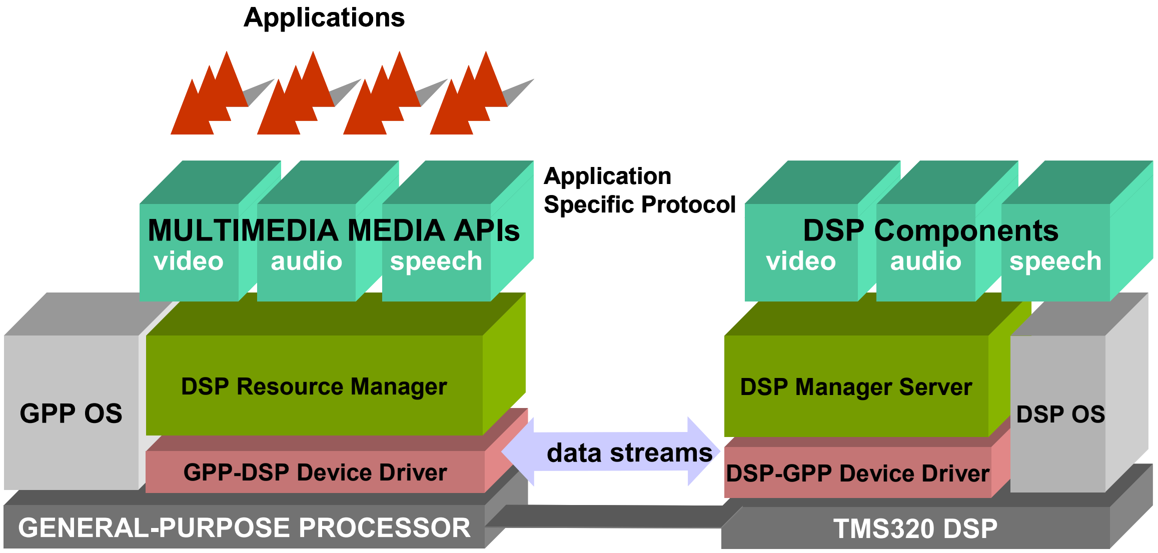

Figure 3: OMAP™ software Architecture

Programmers using the DSP/BIOS Bridge need not become familiar with the minutia of software development on a DSP. Instead, they follow the coding patterns and syntax they are familiar with -- techniques that are similar to the primitives used in operating systems like Symbian OS™ and Windows® CE. For example, if a programmer needs to execute a DSP-based task, he can invoke the DSP/BIOS Bridge on the RISC processor and it will automatically assemble all of the resources required, relieving the designer of these details.

This sort of high-level programming approach can be quite efficient when it is applied to an I/O-intense application like streaming video. For instance, a developer only has to assemble and sequence all of the resources needed for an I/O task or an internal data movement one time. Whenever the task is implemented again, it can be invoked through a high-level command to the DSP/BIOS Bridge, which automatically sets-up, executes and breaks down the task.

Power Down

Another important consideration, addressed in OMAP architecture is power consumption. Because batteries power mobile terminal devices most of the time, power consumption must be squeezed to a minimum to extend the life of the batteries. Unfortunately, video decoding requires significant processor cycles, which consume considerable power. Reducing the processor cycles needed to decode a video bit stream would reduce power consumption.

Not all DSPs are created equally when it comes to video coding/decoding. Some DSP cores, such as the TMS320C55x™ DSP from TI, have instructions and primitive processes that require fewer processor cycles to perform the typical video coding/decoding functions such as inverse discrete cosine (IDCT) and half-pixel interpolations (HPI) on a macroblock. For example, TI's latest-generation DSPs have reduced the cycles for IDCT from 4000 to just 900; while the number of cycles needed for HPIs have fallen from 1500 to a mere 600.

Another factor that affects the power consumed by a mobile device is the size of the displayed image. A large video image means dedicating more processor cycles to the display function, reducing the processor cycles available for other tasks and consuming a great deal of power. For example, the larger common intermediate format (CIF) at 45 frames per second places a processing load of at least 132 million cycles per second on the DSP. In contrast, the smaller quarter common intermediate format (QCIF) at 15 frames per second, which is quite appropriate for many streaming video applications, requires just 12 million DSP cycles per second. In addition, no matter what image format is viewed, TI's advanced DSP-based OMAP architecture consumes much less power than general-purpose RISC processors.

The Foibles of Wireless

Transmitting streaming video over a wireless communications channel is, in and of itself, a challenging proposition. Wireless communications, in general, are prone to inducing errors into the digital bit stream because of interference, weak signals, multipath fading, crowded air waves and any number of other factors. Overcoming these conditions is challenging enough, but today's video compression techniques make matters worse. To fit a streaming video application into the wireless bandwidth that's available today, compression standards like MPEG-4 and H263 have become a way of life. These and other compression techniques help to deliver the video bit stream, but they often work at cross-purposes to the quality of the video image displayed on the terminal device. Fortunately, TI has made strides in this area so that service providers and OEMs need not sacrifice the quality of the image in favor of less bandwidth.
The problem becomes acute because the various compression techniques remove much of the redundancy from a typical video bit stream. The logic behind most video compression methods is that the currently displayed frame is the basis for the frames following it. Video compression

algorithms predict or extrapolate what future frames will look like based on the first frame. This works fine until an error is encountered in the bit stream. Because of the predictive nature of compression techniques, any error in an image could be propagated through successive frames. This uncontrolled propagation of errors can also cause the video decoder to lose synchronization with the video bit stream, leading to a complete failure of the decoding process. For the viewer that means a frozen image on the display screen.

Newer compression standards like MPEG-4 have taken this phenomena into account and a number of techniques are now built into the compression standards to overcome part of the problem. These techniques, which are known as error resilience tools, allow for the detection, containment and concealment of errors in a video bit stream. (See Table 1.) TI has been a leading contributor to the MPEG-4 standard.

Table 1 -- Error Resilience Tools

Tool	**What It Does**
Resynchronization Markers	These are markers that are placed throughout the bit stream, not just at the beginning of a row. This more effectively bounds the extent of an error.
Header Extension Codes (HEC)	When set, an HEC replicates all header information for each video packet, ensuring that a header error does not cause the loss of an entire image.
Data Partitioning	This uses markers to separate motion from texture data so that resynchronization can occur after an error is encountered and all of the data between the two markers is not lost.
Reversible Variable Length Coding	This allows texture data to be decoded in either a forward or a backward direction. If an error causes an inconsistent code word, the decoder can resynchronize on the next marker and then decode in a backward direction, recovering as much data as possible.

Beyond the Standards

Outside the scope of the various error resilience tools that are embodied in the compression standards are several spatial and temporal error concealment techniques. These are more properly thought of as post-processing techniques since they follow the video decoding process. TI has pioneered much of the research in this area and has incorporated the results of its research into the OMAP platform.

For example, one error concealment technique, and a very simple one at that, would replace corrupted macroblocks with the uncorrupted macroblocks from the previous frame. More intelligent concealment techniques can also be implemented; making use of data that has been recovered through the error resilience tools previously mentioned. Typically, error concealment techniques are able to bypass erroneous data to more accurately interpolate the current or next frame.

Of course, everything comes with a price and the price of error concealment is processor cycles. Performing error concealment techniques on a platform without the capabilities of the OMAP architecture would, in all likelihood, degrade the performance of the system because it would siphon off a large portion of the system's processing capabilities.

Further exacerbating this problem is the fact that errors in wireless communications typically occur in bursts. As a result, the processor cycles devoted to error concealment would rise and fall sharply, straining further the resources of the terminal device at those times when the bit stream is prone to extensive errors. Of course, the resources of the dual-processor OMAP platform are extensive and well suited to keeping up with rapid changes in the processing load.

Error concealment also places a strain on a terminal device's I/O channels, because these tools and techniques often require that the processor re-examine past frames to extrapolate more accurately the current frame. When error resilience and/or error concealment techniques are engaged, large blocks of data are flowing back and forth between the DSP processor's external memory and its on-chip RAM. Again, the OMAP architecture shines in this type of environment because of its many DMA channels, which diminish the likelihood that I/O will become a bottleneck.

Fast Forward

Streaming video certainly will be one of several important multimedia applications in 2.5 and 3G. Communications service providers have learned from the personal computer industry that users prefer applications that engage their senses with vibrant colors, movement and audio. These qualities make for a much more enjoyable experience, but they also contribute to the ease-of-use of many applications.

As the wireless market continues to evolve, service providers and terminal device OEMs will realize that users, who are constantly exposed to multimedia applications on their desktops, inevitably will demand more of their mobile devices. Savvy platform designers will place a premium on qualities like flexibility and scalability, because they know that new multimedia applications will continue to strain the resources of wireless devices. Any headroom designed into a device platform today will be quickly absorbed by new multimedia applications tomorrow—applications which will give service providers a competitive advantage in the battle for users as the wireless industry migrates to 2.5G and 3G.

The TI OMAP platform is based on a highly extensible architecture that can be expanded with application-specific processing capabilities and additional I/O so that even the most complicated multimedia applications will execute smoothly and seamlessly.

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.